

BAB IV

PENGUMPULAN DAN PENGOLAHAN DATA

Pada bab ini akan menjelaskan tentang tinjauan perusahaan, pengumpulan pendataan serta pengolahan data.

4.1 Tinjauan Umum Perusahaan

4.1.1 Deskripsi Permasalahan

PT. Jatim Es Tube-Malang (atlas) adalah perusahaan asal Malaysia yang bergerak dibidang es kristal dan memiliki 25 anak perusahaan di Indonesia salah satunya berada di Kota Malang yang terletak di Jl. Simpang Janti Barat 1, Kecamatan Sukun. PT. Jatim Es Tube-Malang (atlas) menjadi salah satu kompetitor dan distributor terbesar dalam bidangnya di wilayah Kota Malang. Perusahaan yang berdiri pada tanggal 18 April 2014 ini memiliki 2 varian produk yaitu es *tube* dan es serut dengan kemasan berat yang berbeda mulai dari 5 kg, 10 kg, dan 20 kg. Proses distribusi Es Kristal yaitu ke seluruh wilayah Kabupaten dan Kota Malang. Sistem distribusi yang dilakukan dengan cara mengisi penuh muatan pada setiap kendaraan berdasarkan permintaan pelanggan dengan muatan es *tube* maupun es serut.

Atlas telah mendistribusikan produknya lebih dari 500 pelanggan yang tersebar di wilayah Kabupaten dan Kota Malang. Pada penelitian ini, hanya di fokuskan pada salah satu kecamatan di wilayah Kota Malang, yaitu Kecamatan Klojen. Banyaknya usaha yang menjual minuman siap saji baik warung kopi sampai pelanggan kecil lainnya, bahkan hingga dari kelas restoran, menjadikan permintaan Es Kristal semakin tinggi. Permintaan Es Kristal yang tinggi menjadikan sebuah kesempatan bagi perusahaan untuk memproduksi dalam jumlah besar. Es Kristal ini lebih praktis dan higienis untuk digunakan.

4.2 Pengumpulan Data

Pada sub bab ini akan menjelaskan tentang pengumpulan data berupa daerah pelanggan, dan rute awal distribusi, data jumlah permintaan Es Kristal dari tiap pelanggan, rute awal, armada pengiriman, data biaya dan konsumsi bahan bakar.

4.2.1 Data Jumlah Permintaan Es kristal dari Tiap Pelanggan

Pengumpulan data ini digunakan untuk menentukan *order size* setiap pelanggan agar dapat disesuaikan dengan kapasitas angkut kendaraan pada saat proses pendistribusian produk, yang selanjutnya akan dilakukan perhitungan dengan metode usulan. Kecamatan Klojen merupakan daerah yang akan dilakukan perbaikan. Adapun dibawah ini adalah data pelanggan yang menjadi tujuan distribusi PT. Jatim Es Tube-Malang (atlas) yaitu sebagai berikut:

Tabel 4.1 Truk 1 dengan kode operasi OO

No.	Customer	Demand	No.	Customer	Demand
1	Pelanggan 1	35	20	Pelanggan 20	70
2	Pelanggan 2	30	21	Pelanggan 21	50
3	Pelanggan 3	90	22	Pelanggan 22	80
4	Pelanggan 4	80	23	Pelanggan 23	20
5	Pelanggan 5	25	24	Pelanggan 24	40
6	Pelanggan 6	65	25	Pelanggan 25	90
7	Pelanggan 7	170	26	Pelanggan 26	10
8	Pelanggan 8	175	27	Pelanggan 27	10
9	Pelanggan 9	50	28	Pelanggan 28	10
10	Pelanggan 10	30	29	Pelanggan 29	20
11	Pelanggan 11	120	30	Pelanggan 30	10
12	Pelanggan 12	10	31	Pelanggan 31	50
13	Pelanggan 13	5	32	Pelanggan 32	10
14	Pelanggan 14	20	33	Pelanggan 33	50
15	Pelanggan 15	10	34	Pelanggan 34	10

16	Pelanggan 16	60	35	Pelanggan 35	20
17	Pelanggan 17	125	36	Pelanggan 36	10
18	Pelanggan 18	15	37	Pelanggan 37	10
19	Pelanggan 19	35	38	Pelanggan 38	10

Tabel 4.2 Truk 2 dengan kode operasi MM

No.	Customer	Demand	No.	Customer	Demand
1	Pelanggan 39	70	11	Pelanggan 49	45
2	Pelanggan 40	110	12	Pelanggan 50	110
3	Pelanggan 41	200	13	Pelanggan 51	215
4	Pelanggan 42	240	14	Pelanggan 52	90
5	Pelanggan 43	520	15	Pelanggan 53	55
6	Pelanggan 44	235	16	Pelanggan 54	40
7	Pelanggan 45	200	17	Pelanggan 55	80
8	Pelanggan 46	265	18	Pelanggan 56	50
9	Pelanggan 47	200	19	Pelanggan 57	40
10	Pelanggan 48	25			

Tabel 4.3 Truk 3 dengan kode operasi BB

No.	Customer	Demand	No.	Customer	Demand
1	Pelanggan 58	130	12	Pelanggan 69	100
2	Pelanggan 59	60	13	Pelanggan 70	50
3	Pelanggan 60	60	14	Pelanggan 71	35
4	Pelanggan 61	30	15	Pelanggan 72	10
5	Pelanggan 62	60	16	Pelanggan 73	80
6	Pelanggan 63	80	17	Pelanggan 74	100
7	Pelanggan 64	10	18	Pelanggan 75	260
8	Pelanggan 65	60	19	Pelanggan 76	180
9	Pelanggan 66	25	20	Pelanggan 77	850

10	Pelanggan 67	20	21	Pelanggan 78	235
11	Pelanggan 68	50	22	Pelanggan 79	30

4.2.2 Jarak Distribusi Awal Kecamatan Klojen

Pada wilayah Kecamatan Klojen terdapat 3 rute yang dimiliki PT. Jatim Es Tube-Malang (Atlas) yaitu rute Rijal (OO), dengan jarak 140,74 km serta rute Pak Hari (MM), dengan 107,16 km dan rute Delita (BB) dengan 168,08. Berikut adalah rute awal setiap armada truk di wilayah Kecamatan Klojen yang telah didapatkan peneliti.

Tabel 4.4 rute awal perusahaan

Rute	Demand (kg)	Jarak (km)	Biaya
Truk 1 A-P1-P2-P3-P4-P5-P6-P7-P8-P9-P10-P11-P12-P13-P14-P15-P16-P17-P18-P19-P20-P21-P22-P23-P24-P25-P26-P27-P28-P29-P30-P31-P32-P33-P34-P35-P36-P37-P38-A	1730	140,74	Rp 102.500
Truk 2 A- P39-P40-P41-P42-P43-P44-P45-P46-P47-P48-P49-P50-P51-P52-P53-P54-P55-P56-P57-A	2790	107,16	Rp 78.000
Truk 3 A- P58-P59-P60-P61-P62-P63-P64-P65-P66-P67-P68-P69-P70-P71-P72-P73-P74-P75-P76-P77-P78-P79-A	2515	168,08	Rp 122.500
Total	7035	415,98	Rp 303.000

Keterangan:

A = Agen/perusahaan

P1,2,...n = Pelanggan ke 1 sampai pelanggan ke-n

4.2.3 Armada Pengiriman

Dalam melakukan pengiriman es kristal, PT. Jatim Es Tube-Malang (atlas) menggunakan kendaraan berupa truk engkel. Jumlah kendaraan, jenis kendaraan, dan kapasitas kendaraan adalah sebagai berikut:

- Jumlah kendaraan : 3 Truk
- Jenis kendaraan dan kapasitas:
 - Hino 110 SD.
 - Kapasitas : 3.500 Kg.
 - Bahan bakar : Solar
 - Jarak tempuh/liter : 7 km/liter
 - Harga solar : Rp. 5.150/liter

4.2.4 Data Biaya

Biaya-biaya yang dikeluarkan selama proses pendistribusian es kristal yaitu meliputi biaya bahan bakar. Biaya tersebut merupakan biaya yang biasanya dikeluarkan perusahaan setiap bulannya dalam pendistribusian es kristal. Berikut ini merupakan biaya transportasi, yaitu:

4.2.5 Konsumsi Bahan Bakar

Konsumsi bahan bakar yang dikeluarkan perusahaan untuk rute OO Rp 102.500/hari dengan kebutuhan solar 20.11 liter/hari, serta rute MM Rp 78.000/hari dengan kebutuhan 15,31 liter/hari dan rute BB Rp 122.500/hari dengan kebutuhan 24.01 liter/hari. Harga solar Rp 5.150/liter, maka perusahaan membutuhkan Rp 303.000/hari dengan kebutuhan solar 59,43 liter/hari untuk rute di Kecamatan Klojen.

4.2.6 Pengukuran Jarak dari Perusahaan ke Pelanggan dan Antar Pelanggan

Pengukuran dilakukan untuk mengetahui jarak dari perusahaan ke pelanggan dan antar pelanggan yang akan dikunjungi. Pengukuran dilakukan dengan menggunakan *google maps*.

Langkah-langkah pengukuran jarak dengan menggunakan *google maps*:

1. Buka *google maps*.
2. Tandai lokasi perusahaan dan pelanggan-pelanggan yang akan dikunjungi.
3. Kemudian ukur jarak dari perusahaan ke pelanggan dan antar pelanggan yang akan dikunjungi.

4.3 Pengolaha Data

4.3.1 Algoritma usulan untuk menentukan rute layak pada study kasus TSPTW

1. Melakukan penugasan dan penentuan solusi awal dengan menggunakan algoritma saving matrix dengan kendala kapasitas dan waktu tempuh.

Penentuan rute awal untuk masing-masing kendaraan dengan menggunakan metode *saving matrix*.

Setelah melakukan proses perhitungan matriks penghematan dan telah didapatkan hasilnya maka proses selanjutnya yang harus dilakukan adalah mengalokasikan konsumen-konsumen ke sebuah rute/kendaraan menugaskan konsumen pada sebuah rute. Dalam melakukan mengalokasikan konsumen pada rute/kendaraan harus memperhatikan matriks penghematan, dengan mencari matriks penghematan tertinggi. Kemudian selanjutnya mempertimbangkan kapasitas kendaraan dan jumlah permintaan dari masing-masing konsumen, selain itu pengalokasian konsumen ke sebuah rute/kendaraan harus bisa memaksimalkan penghematan. Pencarian solusi dilakukan dengan prosedur iteratif dengan langkah-langkah sebagai berikut:

- ✓ Pada tahap 1 : Tiap konsumen dialokasikan pada truk/rute yang berbeda-beda/terpisah.
- ✓ Pada tahap 2 : Konsumen selanjutnya dapat digabungkan pada satu truk/rute dengan didasarkan pada penghematan yang paling tinggi yang bisa diperoleh. Selanjutnya dilakukan pengecekan apakah pengkombinasian tersebut layak atau tidak. Dikatakan layak jika total pengiriman yang harus dilalui melalui rute tersebut tidak melebihi kapasitas kendaraan. Jika tidak

layak maka dilakukan penghematan tertinggi berikutnya sampai jumlah pengiriman dinyatakan cukup.

- ✓ Pada tahap 3 : Untuk konsumen lain yang belum dialokasikan maka kembali dilakukan pengalokasian sesuai dengan tahap 2.

Untuk memperjelas maksud dari ketiga langkah diatas maka berikut dapat dilihat proses penentuan alokasi pelanggan ke setiap alat angkut secara singkat.

2. Penentuan solusi optimal dengan menggunakan algoritma tabu search.

Tabu Search adalah salah satu metaheuristik yang sudah cukup lama ditemukan. Beberapa karakteristik dari Tabu Search:

1. Metode descent tradisional tidak menerima pergerakan yang tidak memperbaiki solusi, TS bisa menerima solusi yang lebih buruk.
2. Kebanyakan metaheuristik adalah melalui proses random, sedangkan TS lebih deterministic, tidak ada pembangkitan bilangan random.

Ada beberapa konsep yang harus dipahami sebelum kita menggunakan TS. Berikut adalah beberapa konsep yang digunakan dalam TS.

1. Tabu list : adalah daftar yang berisi solusi yang sudah pernah dicoba. Tabu list sering disebut sebagai memori jangka pendek. Memori tersebut dibangkitkan dari beberapa iterasi. Ukuran tabu list harus dibatasi untuk tidak menampung solusi terlalu banyak.
2. Aspiration criteria: kriteria ini dimaksudkan untuk mengganti solusi terbaik yang sudah pernah dihasilkan jika ada solusi baru yang lebih baik.
3. Aspiration criterion memungkinkan pergerakan yang tabu asal menuju solusi yang lebih baik.
4. Intensifikasi (memori jangka menengah): memori jangka menengah yang menyimpan sejumlah solusi yang berkualitas dari hasil selama proses pencarian.

5. Diversifikasi (memori jangka panjang): memori jangka panjang menyimpan informasi tentang kandidat solusi yang pernah dikunjungi.
6. Tabu tenure: durasi suatu pergerakan tetap dianggap tabu. Tabu tenure termasuk fitur penting karena menentukan seberapa ketat pencarian local dilakukan. Ukuran tabu tenure t dipilih bergantung pada ukuran problem. Misalkan problem berukuran n maka $t = 1/2 n$. Ukurannya biasanya jumlah iterasi.

Algoritma *Tabu Search*

1. Bangkitkan solusi awal, S
2. Buat tabu list, memori jangka menengah, memori jangka panjang,
Sementara kriteria penghentian belum tercapai, lakukan:
Temukan solusi terbaik sejauh ini, S^*
 $S = S^*$
Perbaiki tabu list, aspiration criteria, memori jangka menengah dan memori jangka panjang
Jika memenuhi kriteria intensifikasi lakukan intensifikasi
Jika memenuhi kriteria diversifikasi lakukan diversifikasi stop.

4.3.1.1 Algoritma Saving Matrix

Input : jumlah node, jumlah kendaraan, matrix waktu, demand, dan kapasitas kendaraan.

Output : rute terbaik, total jarak dan kapasitas kendaraan.

Langkah 1 pembuatan matrix penghematan

Untuk setiap node, hitung matrix penghematan waktu tempuh dengan

$$\text{rumus: } S(x,y) = \text{Dist}(DC,x) + \text{Dist}(DC,y) - \text{Dist}(x,y)$$

Langkah 2 pencarian rute layak dengan batasan kapasitas kendaraan untuk setiap kendaraan cari nilai tertinggi pertama dari seluruh matrix

Langkah 3 memastikan bahwa rute dimulai dari pabrik dan kembali ke pabrik

Untuk setiap rute : perbarui rute dengan ketentuan sebagai berikut rute $[1 \text{ rute} + 1 \ 1]$

4.3.1.2 Algoritma untuk mengevaluasi rute

Memastikan kelayakan solusi dengan batasan time windows (jam buka dan jam tutup pabrik).

Untuk setiap rute : $i=1$: jumlah rute $- 1$

1. $J = i + 1$
2. Hitung waktu tempuh $wts = wts + tx(i,j)$
3. Jika $wts > tw$, maka rute di stop dan dilanjutkan ke kendaraan selanjutnya

4.3.1.3 Algoritma Tabu Search

Langkah-langkah TS untuk TSP

1. Inisialisasi
 - Input: jumlah kota n , matrix jarak, maksimum iterasi
 - Set tabu tenure: jumlah iterasi bahwa untuk berpindah tetap tabu, $t = \sqrt{n}$.
 - Inisialisasi tabu list = []
2. Gunakan Greedy procedure untuk menemukan solusi terbaik x^* dengan jarak total z^* :
 - misalkan x , rute sekarang
 - tetapkan $x^* = x$, rute terbaik
 - tetapkan $z^* = z(x)$ (jarak total)
3. Lakukan iterasi

Sementara kriteria pemberhentian belum dicapai, lakukan

 - Identifikasi Neighborhood set dengan menciptakan beberapa solusi baru. Bisa menggunakan swap (tukar), slide (geser) atau flip (balik urutan) dari solusi x , setiap x jumlah X mempunyai Neighborhood $N(x)X$.

4. Pilih pergerakan yang terbaik dari $N(x)$ dengan memilih solusi dengan minimum $Z(x)$, x menjadi x' jumlah $N(x)$ dan nilai fungsi obyektif menjadi $z(x')$ dan masukkan ke dalam tabu list.
 - cek apakah x' tabu:
 - jika tidak, ke langkah 5
 - jika ya, step 6
5. Lakukan pertukaran:
 - $x=x'$, $z(x)=z(x')$.
 - jika $z(x) < z^*$ maka $z^*=z(x)$, $x^*=x$
6. Cek apakah x' memenuhi kriteria Aspirasi
 - jika ya, ke langkah 5
 - jika tidak, cek neighborhood set
7. Catat tabu untuk perpindahan sekarang dalam tabu list, update tabu list, dengan cara mengubah isi tabu list sebagai berikut:
 - Jika solusi tidak tabu, tambahkan solusi pada tabu list dengan mengubah posisi kota dari 0 ke t
 - Untuk contoh jika problem mempunyai 5 kota dan solusi yang dihasilkan (swap city 4 & city 3) tidak tabu maka tabu list harusnya [000t0] pada iterasi berikutnya tabu tenure turun 1
 - Ulangi untuk sejumlah iterasi

Hasil: solution x^* dengan jarak total minimum z^* .

4.3.2 Pembuatan Syntax Matlab

Syntax Saving Matrix

```
function [rute]=savmat(n,v,tx,D,C)
%n=size(tx,2)-1  %n adalah jumlah node yang harus dikunjungi
x=zeros(n,n)
for i=1:n
    for j=i+1:n
        if j==i
            x(i,j)=0
        else
```

```

        xij=tx(1,i+1)+tx(1,j+1)-tx(i+1,j+1)
        x(i,j)=xij
    end
end
mx=zeros(1,n)
for i=2:n
    mx(1,i)=max(x(:,i))
end

rute=zeros(v,n)
for r=1:v
    bbn=0
    if size(find(mx==0),2)~=n
        i=1
        while i<=n
            if size(find(mx==0),2)~=n
                a=max(mx)
                node=find(mx==a)
                %cek kapasitas kendaraan
                bbn=bbn+D(1,node(1,1))
                if bbn<=C
                    rute(r,i)=node(1,1)
                    mx(1,node(1,1))=0
                end
            else
                rute(r,i)=1
                break
            end
            i=i+1
        end
    else
        break
    end
end
jartot=zeros(v,1)
for a=1:v
    r=rute(a,:)
    js=0
    b=find(rute(a,:)~=0)
    for i=1:size(b,2)-1
        j=i+1
        js=js+tx(r(1,i),r(1,j))
    end
    jartot(a)=js
end

%Memastikan bahwa rute dimulai dari pabrik dan kembali ke pabrik
for r=1:size(rute,1)
    if rute(r,1)~=0
        rs=find(rute(r,:)~=0)

```

```

        rute(r,1:size(rs,2)+2)=[1 rute(r,rs)+1 1]
    end
end

%Memastikan kelayakan solusi
r=1
while r<=size(rute,1)-1
    routes=rute(r,find(rute(r,:)~=0))
    %rutet=zeros(size(rute,1)-r+1,size(rute,2))
    rutet=zeros(size(rute))
    wts=0
    for i=1:size(routes,2)-1
        j=i+1
        %Hitung waktu tempuh
        wts=wts+tx(routes(i),routes(j))
        if wts>tw(routes(j),2)
            %Update rute
            wts=wts-tx(routes(i),routes(j))
            if wts+tx(routes(i),1)<=tw(routes(1),2)
                rutet(1,1:i+1)=[routes(1:i) 1]
                a=routes(r,i+1:size(routes,2)-1)
                b=rute(r+1,2:size(find(rute(r+1,:)~=0),2)-1)
                c=[1 a b 1]
                if size(a,2)+size(b,2)+2<=size(rutet,2)
                    rutet(2,1:size(a,2)+size(b,2)+2)=[1 a b 1]
                    rutet(3,:)=rute(3,:)
                else
                    d=c(1:size(rutet,2)-1)
                    rutet(2,:)=d 1]
                    rutet(3,1:size(c,2)-(size(rutet(2),2)-1))=[1
c(size(d,2),:)]
                end
            else
                rutet(r,:)=routes(1:i-1) 1]
                a=routes(r,i+1:size(routes,2)-1)
                b=rute(r+1,2:size(find(rute(r+1,:)~=0),2)-1)
                c=[1 a b 1]
                if size(a,2)+size(b,2)+2<=size(rutet,2)
                    rutet(2,1:size(a,2)+size(b,2)+2)=[1 a b 1]
                    rutet(3,:)=rute(3,:)
                else
                    d=c(1:size(rutet,2)-1)
                    rutet(2,:)=d 1]
                    rutet(3,1:size(c,2)-(size(rutet(2),2)-1))=[1
c(size(d,2),:)]
                end
            end
            break
        end
    end
    rute(r:size(rute,1),:)=rutet(r:size(rute,1),:)
    r=r+1
end

```

end

Syntax Tabu Search

```
function [BestSol,BestJarak]=tabusearch_budi(rute,d,MaxIt)
clc

n=size(find(rute~=0),2);
ActionList=permaction(n-1); %daftaraksi
nAction=numel(ActionList); %jumlahaksi
TL=round(0.5*nAction); %Tabulength

%%initialization
%Create Empty Individual Structure
empty_individual.Rute=[];
empty_individual.Jarak=[];
%bangkitkan solusi awal
sol=empty_individual
sol.Rute=rute(1,find(rute~=0))
sol.Jarak=jartot([1 sol.Rute+1 1],d)
%jarak total
BestSol=sol;

%Array untuk menyimpan biaya terkecil
BestBiaya=zeros(MaxIt,1);
%vektor untuk menyimpan jumlah tabu
TC=zeros(nAction,1);

%%tabu search main loop
for it=1:MaxIt
    bestnewsol.Jarak=inf;
    for i=1:nAction
        if TC(i)==0
            newsol.Rute=DoAction(sol.Rute,ActionList{i});
            newsol.Jarak=jartot([1 newsol.Rute+1 1],d);
            %newsol.Biaya=jartot([newsol.Rute...
            %newsol.Rute(i)],d);
            newsol.ActionIndex=i;
            if newsol.Jarak<=bestnewsol.Jarak
                bestnewsol=newsol;
                bestnewsol.ActionIndex=newsol.ActionIndex;
            end
        end
    end
end

sol=bestnewsol;

%UpdateTabuList
for i=1:nAction
    if i==bestnewsol.ActionIndex
```

```

        TC(i)=TL; %Add to tabu list
    else
        TC(i)=max(TC(i)-1,0); %reduce tabu counter
    end
end

%update best solution ever found
if sol.Jarak<=BestSol.Jarak
    BestSol=sol;
end

%save best biaya ever found
BestJarak(it)=BestSol.Jarak;

%if global minimum is reached
if BestJarak(it)==0
    break;
end
end
BestSol.Rute=[1 BestSol.Rute+1 1]
BestJarak=BestJarak(it);

doAction

function q=DoAction(p,a)
switch a(1)
case 1
    q=DoSwap(p,a(2),a(3));
case 2
    q=DoFlip(p,a(1),a(3));
case 3
    q=DoSlide(p,a(2),a(3));
end

Permaction

function ActionList=permaction(n)

nSwap=n*(n-1)/2;
nFlip=n*(n-1)/2;
nSlide=n^2;
nAction=nSwap+nFlip+nSlide;
ActionList=cell(nAction,1);

c=0;

%Add SWAP

```

```

for i=1:n-1
    for j=i+1:n
        c=c+1;
        ActionList{c}=[1 i j];
    end
end

%add flip
for i=1:n-1
    for j=i+1:n
        if abs(i-j)>2
            c=c+1
            ActionList{c}=[2 i j];
        end
    end
end

%add slide
for i=1:n
    for j=1:n
        if abs(i-j)>1
            c=c+1;
            ActionList{c}=[3 i j];
        end
    end
end

ActionList=ActionList(1:c);
End

```

4.3.3 Validasi Model

Pada tahapan ini akan dilakukan perbandingan antara perhitungan manual dengan perhitungan menggunakan matlab yang bertujuan untuk mengetahui apakah model formulasi layak digunakan atau tidak. Model yang akan di uji coba menggunakan matrix 10 x 10 dari 79 x 79 data penelitian skripsi. Sedangkan untuk hasil perhitungan keseluruhan data terdapat di lampiran. Validasi model ini dilakukan dengan menggunakan percobaan 10 pelanggan dengan perhitungan manual dan perhitungan menggunakan bantuan MATLAB.

Adapun beberapa langkah sebelum perbandingan antara perhitungan manual dan perhitungan menggunakan bantuan MATLAB.

Langkah 1 Penentuan solusi awal dengan algoritma *saving matrix*

Langkah 1 a membuat *matrix* penghematan

Langkah 1 b menentukan rute awal sesuai dengan kapasitas truk

Memilih pelanggan pertama dengan jumlah penghematan tertinggi pertama. P8 dipilih terlebih dahulu karena memiliki *matrix* penghematan tertinggi pertama dengan demand sebesar 175 kg. Kemudian mencari penghematan *matrix* terbesar kedua, dipilihlah kode pelanggan P10 dengan demand sebesar 30 kg dan seterusnya sampai kurang dari sama dengan 1000 kg atau memenuhi kapasitas truk.

Contoh perhitungan.

$$P8 = 175\text{kg} \leq 1000$$

$$P8+P10 = 175+30 = 205 \leq 1000$$

$$P8+P10+P9+P7+P6+P5+P4+P2+P3+P1 = 175+30+50+170+65+25+80+30+90+35 \\ = 750 \text{ kg}$$

Langkah ke 2 mengevaluasi dengan menggunakan Algoritma Tabu Search

Sebagai solusi terbaik untuk tahap awal. Rute awal ditentukan dengan menggunakan ketetanggaan terdekat dan diperoleh rute awal perjalanan A-8-10-9-7-6-5-4-2-3-1-A. Rute awal ini masuk dalam Tabu List pada iterasi 0 sekaligus sebagai solusi awal untuk jarak tempuh dan waktu perjalanan.

Menentukan iterasi selanjutnya dan mencari solusi alternatif. Solusi alternatif diperoleh dengan *neighborhood search* menggunakan aturan kombinasi. Penyelesaian TSP untuk mendapatkan jarak tempuh dan waktu perjalanan yang optimal digunakan dengan cara menukar 2 titik atau menukar posisi 2 jarak/waktu secara berurutan. Untuk mencari jumlah kombinasi dari permasalahan tersebut dengan kondisi perjalanan yang dilakukan dengan mencari rute optimal dan setiap pos hanya boleh dikunjungi tepat satu kali. Berikut adalah proses pencarian jalur alternatif untuk iterasi-iterasi tersebut:

Iterasi 1:

Pencarian jalur alternative untuk jarak tempuh dan waktu tempuh.

Jalur solusi awal :

A-8-9-10-7-6-5-4-2-3-1-A dengan jarak tempuh adalah 25,6 km dan waktu perjalanan 38,4 menit

Tabu List iterasi 1 : A-8-10-9-7-6-5-4-2-3-1-A

Tabel 4.6 Iterasi manual

Iterasi	Rute perjalanan	Jarak tempuh (km)	Waktu perjalan (menit)
Ke-1	A-9-8-10-7-6-5-4-2-3- 1-A	25,7	38,55
Ke-2	A-10-9-8-7-6-5-4-2-3- 1-A	23,24	34,86
Ke-n	A-1-2-6-8-7-9-10-5-4- 3-A	18,44	27,66

Pada iterasi ke-2 ini diperoleh nilai terbaik yaitu 23,24 km untuk jarak tempuh dan 34,86 menit untuk waktu tempuh. Setelah dilakukan perhitungan sebanyak 20 iterasi, maka diperoleh jarak tempuh dan waktu perjalanan *minimum* pada setiap iterasi tersebut. Jarak tempuh dan waktu perjalanan *minimum* yang diperoleh dari setiap iterasi disimpan ke dalam *Tabu List*.

Langkah ke 4 adalah perbandingan antara perhitungan manual dengan perhitungan menggunakan MATLAB.

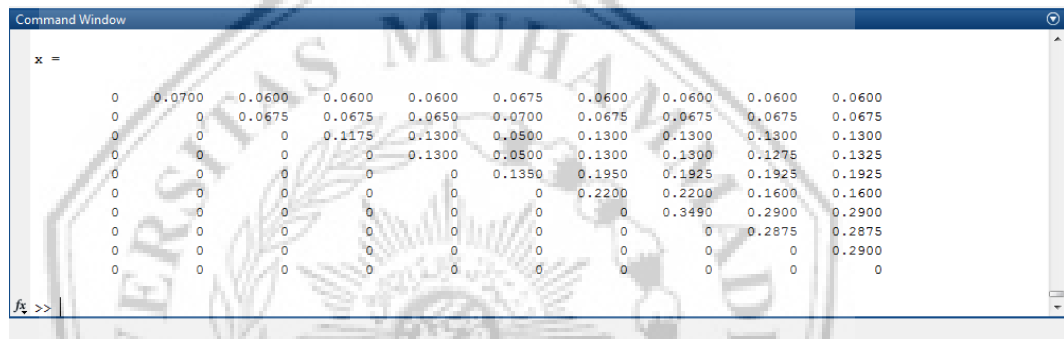
Validasi 1:

Perbandingan hasil perhitungan matrix penghematan secara manual dan menggunakan matlab. Perhitungan menggunakan matlab dianggap layak jika hasil dari perhitungan manual sama dengan dengan hasil perhitungan menggunakan matlab.

Tabel 4.7 perhitungan manual matrix penghematan 10 x 10

	P1	P2	P3	P4	P5	P6	P7	P8	P9	P10
P1	0	0.07	0.06	0.06	0.06	0.0675	0.06	0.06	0.06	0.06
P2	0	0	0.0675	0.0675	0.065	0.07	0.0675	0.0675	0.0675	0.0675

P3	0	0	0	0.1175	0.13	0.05	0.13	0.13	0.13	0.13
P4	0	0	0	0	0.13	0.05	0.13	0.13	0.1275	0.1325
P5	0	0	0	0	0	0.135	0.195	0.1925	0.1925	0.1925
P6	0	0	0	0	0	0	0.22	0.22	0.16	0.16
P7	0	0	0	0	0	0	0	0.349	0.29	0.29
P8	0	0	0	0	0	0	0	0	0.2875	0.2875
P9	0	0	0	0	0	0	0	0	0	0.29
P10	0	0	0	0	0	0	0	0	0	0



```

Command Window
x =
    0    0.0700    0.0600    0.0600    0.0600    0.0675    0.0600    0.0600    0.0600    0.0600
    0         0    0.0675    0.0675    0.0650    0.0700    0.0675    0.0675    0.0675    0.0675
    0         0         0    0.1175    0.1300    0.0500    0.1300    0.1300    0.1300    0.1300
    0         0         0         0    0.1300    0.0500    0.1300    0.1300    0.1275    0.1325
    0         0         0         0         0    0.1350    0.1950    0.1925    0.1925    0.1925
    0         0         0         0         0         0    0.2200    0.2200    0.1600    0.1600
    0         0         0         0         0         0         0    0.3490    0.2900    0.2900
    0         0         0         0         0         0         0         0    0.2875    0.2875
    0         0         0         0         0         0         0         0         0    0.2900
    0         0         0         0         0         0         0         0         0         0

```

Gambar 4.1 perhitungan matrix penghematan 10 x 10 menggunakan matlab

Berdasarkan table diatas, dapat dilihat bahwa hasil dari perhitungan manual matrix penghematan 10 x 10 pada matrix P1 ke P2 = 0.07 sedangkan hasil perhitungan menggunakan matlab juga menunjukkan 0.0700. Maka dapat disimpulkan bahwa perhitungan menggunakan matlab layak untuk digunakan.

Validasi 2:

Perbandingan selanjutnya yaitu pada hasil urutan rute pengiriman dengan tahapan awal membuat matrix penghematan. Untuk tahapan mencari rute sudah dijelaskan diatas.

Perhitungan menggunakan matlab dianggap layak jika hasil dari perhitungan manual sama dengan dengan hasil perhitungan menggunakan matlab.

Hasil dari perhitungan manual.

A-8-10-9-7-6-5-4-2-3-1-A

Hasil perhitungan matlab



```

rute =

     1     9    11    10     8     7     6     5     3     4     2     1
     0     0     0     0     0     0     0     0     0     0     0     0
     0     0     0     0     0     0     0     0     0     0     0     0

fx >>
  
```

Gambar 4.2 Hasil perhitungan dengan bantuan software matlab

Berdasarkan hasil perhitungan diatas, maka dapat disimpulkan bahwa fungsi matlab layak untuk digunakan untuk data yang lebih besar.

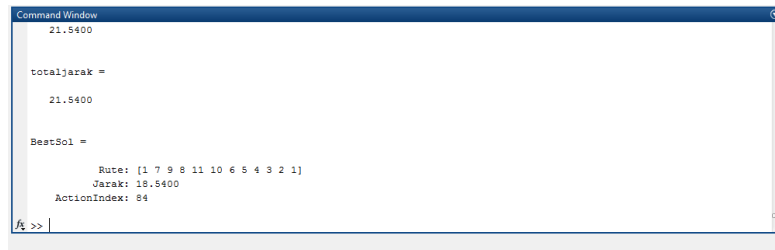
Validasi 3:

Perbandingan hasil iterasi dengan menggunakan algoritma tabu search secara manual dan menggunakan matlab dengan asumsi bahwa batasan iterasi maksimal adalah 20 iterasi.

Tabel 4.8 Hasil perhitungan manual

Iterasi	Rute	Jarak/km	Waktu/menit
Ke-20	A-6-8-7-10-9-5-4-3-2-1-A	18,54	27,66

Sedangkan hasil menggunakan formulasi algoritma tabu search adalah sebagai berikut:



```

Command Window
21.5400

totaljarak =

21.5400

BestSol =

Route: [1 7 9 8 11 10 6 5 4 3 2 1]
Jarak: 18.5400
ActionIndex: 84
>>

```

Gambar 4.3 Hasil sequencing menggunakan tabu search dengan bantuan software

Berdasarkan hasil perhitungan diatas, maka dapat disimpulkan bahwa fungsi matlab layak digunakan untuk data yang lebih besar.

4.4 Hasil perhitungan menggunakan bantuan *software* matlab

Perhitungan menggunakan bantuan *software* matlab ini menggunakan data pada tabel 4.1 4.2 4.3 dengan parameter *saving matrix*.

jumlah kendaraan : 3

jumlah pelanggan : 79

kapasitas kendaraan : 3500 kg

Serta dengan parameter *tabu search* dengan iterasi sebesar 100 didapatkan hasil sebagai berikut

Truk 1

Pabrik – 46 – 45 – 47 – 44 – 78 – 41 – 42 – 43 – 77 – 76 – 75 – Pabrik.

Dengan total demand sebesar 3.385 kg

Truk 2

Pabrik – 48 – 33 – 32 – 35 – 34 – 74 – 31 – 36 – 79 – 30 – 29 – 73 – 28 – 40 – 27 – 26 – 38 – 39 – 25 – 37 – 24 – 64 – 63 – 49 – 72 – 62 – 65 – 67 – 66 – 60 – 59 – 58 – 61 – 68 – 70 – 69 – 57 – 16 – 56 – 55 – 17 – 71 – 22 – 21 – 23 – 20 – 8 – 18

– 19 – 54 – 10 – 9 – 15 – 13 – 11 – 12 – 14 – 7 – 53 – 6 – 5 – 4 – 52 – 51 –
Pabrik.

Dengan total demand sebesar 3.385 kg

Truk 3

Pabrik – 50 – 2 – 3 – 1 – Pabrik.

Dengan total demand sebesar 265 kg

Hasil running matlab sequencing menggunakan metode tabu search

Truk 1

Pabrik – 76 – 78 – 47 – 45 – 44 – 43 – 42 – 46 – 41 – 77 – 75 – Pabrik.

Truk 2

Pabrik – 52 – 33 – 32 – 35 – 34 – 31 – 30 – 29 – 15 – 14 – 13 – 11 – 79 – 74 – 73
– 72 – 71 – 48 – 40 – 37 – 27 – 26 – 23 – 39 – 38 – 36 – 28 – 24 – 25 – 22 – 20 –
19 – 18 – 17 – 16 – 12 – 10 – 9 – 8 – 7 – 6 – 54 – 70 – 64 – 63 – 62 – 61 – 67 –
66 – 60 – 59 – 58 – 56 – 69 – 68 – 65 – 57 – 55 – 53 – 49 – 5 – 21 – 4 – 51 –
Pabrik.

Truk 3

Pabrik – 50 – 2 – 3 – 1 – Pabrik.